

Abstract

A parallelized or array method is developed for the generation of Reed Solomon parity bytes which utilizes multiple digital logic operations or computer instructions implemented using digital logic. At least one of the operations or instructions used performs the following combinations of steps: a) provide an operand representing N feedback terms where N is greater than one, b) computation of N by M Galios Field polynomial multiplications where M is greater than one, and c) computation of $(N-1)$ by M Galios Field additions producing M result bytes. In this case the result bytes are used to modify the Reed Solomon parity bytes in either a separate operation or instruction or as part of the same operation.

A parallelized or array method is also developed for the generation of Reed Solomon syndrome bytes which utilizes multiple digital logic operations or computer instructions implemented using digital logic. At least one of the operations or instructions performs the following combinations of steps: a) provide an operand representing N data terms where N is one or greater, b) provide an operand representing M incoming Reed Solomon syndrome bytes where M is greater than one, c) computation of N by M Galios Field polynomial multiplications, d) computation of N by M Galios Field additions producing M modified Reed Solomon syndrome bytes.

The values of N and M may be selected to match the word width of the candidate MIPS microprocessor which is 32 bits or four bytes. When N and M are both have the value of four, sixteen Galios Field polynomial multiplications may be computed concurrently or sequentially in a pipeline. Each Galios Field polynomial multiplication utilizes a coefficient delivered from a memory device, which in a preferred embodiment, would be implemented either by a read only memory (ROM), random access memory (RAM) or a register file. The generation of Reed Solomon parity bytes requires several iterations each time using previous modified Reed Solomon parity bytes as incoming Reed Solomon parity bytes. Similarly, the generation of Reed Solomon syndrome bytes requires several iterations each time using previous modified Reed Solomon syndrome bytes as incoming Reed Solomon syndrome bytes.